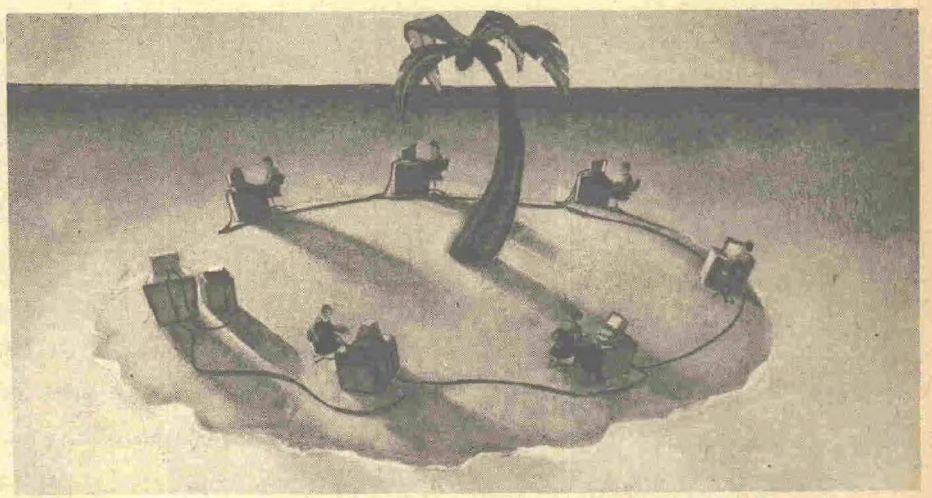
สมศักดิ์ ทองศาสตร์

เทคนิคการเขียนโปรแกรมด้วยฟอกซ์โปร ตอน LAN กับการประยุกต์ใช้งาน

แนะนำวิธีการพัฒนาโปรแกรม เพื่อให้สามารถทำงานในระบบแลนในลักษณะมัลติยูสเซอร์ได้



ถึงวันนี้คงไม่มีใครกล้าปฏิเสธระบบเครือข่ายท้องถิ่น หรือ Local Area Network (LAN) กันอย่างแน่นอน หลายองค์กรไม่ว่าจะเป็นบริษัท ห้างร้าน หรือแม้แต่หน่วยงานราชการหลายแห่งได้เริ่มนำระบบแลนเข้ามาประยุกต์ใช้กับการทำงานจนประสบความสำเร็จกันอยู่ทั่วไป อันเป็นผลมาจากคอมพิวเตอร์เป็นสิ่งที่ทุกคนเรียนรู้ได้อย่างไม่ยากนัก ซึ่งต่างกับสมัยก่อนที่เป็นเรื่องเฉพาะสำหรับนักคอมพิวเตอร์จริง ๆ มีการใช้งานง่าย ประกอบกับมีโปรแกรมสำเร็จรูปเฉพาะด้านให้เลือกใช้มากมาย คอมพิวเตอร์จึงเป็นที่นิยมกันอย่างแพร่หลาย เริ่มตั้งแต่งานจัดพิมพ์เอกสาร (Word Processing) สเปรดชีต (Spreadsheet) ไปจนถึงงานจัดเก็บฐานข้อมูล (Database) เป็นต้น

สิ่งที่อาจจะเรียกได้ว่าเป็นความสามารถหรือ feature ที่สำคัญที่สุดของระบบแลนก็เห็นจะเป็นการแบ่งกันใช้ข้อมูล และโปรแกรมในระบบเครือข่าย ซึ่งนักพัฒนาโปรแกรมทั้งมือใหม่และมือเก่าที่เดิมเขียนโปรแกรมสำหรับใช้ในเครื่องที่มีผู้ใช้คนเดียว (Standalone หรือ Single user) มาแล้ว ก็ใฝ่ฝันที่จะเขียนโปรแกรมแบบผู้ใช้หลายคนหรือมัลติยูสเซอร์ (Multiuser) กับเขาบ้าง

บทความในตอนนี้ จึงขอแนะนำวิธีการพัฒนาโปรแกรมเพื่อให้สามารถทำงานในระบบแลนในลักษณะเป็นมัลติยูสเซอร์ได้ โดยจะเน้นที่โปรแกรมฟอกซ์เบสและฟอกซ์โปรเป็นหลัก ซึ่งผู้เขียนพิจารณาแล้วเห็นว่าเป็นที่นิยมในหมู่นักพัฒนาโปรแกรมสำหรับคอมพิวเตอร์ในบ้านเรา

ก่อนอื่นคุณจะต้องติดตั้งโปรแกรมฟอกซ์เบสหรือฟอกซ์โปรที่เป็นเวอร์ชันมัลติยูสเซอร์เสียก่อน จึงจะสามารถเขียนโปรแกรมให้ใช้งานในระบบมัลติยูสเซอร์ได้

เริ่มต้นกันเลย

สิ่งแรกที่จะต้องคำนึงถึงในระบบมัลติยูสเซอร์ก็คือ จะต้องมีผู้ใช้ (User) เพียงคนเดียวเท่านั้นที่สามารถเขียนข้อมูลลงไฟล์หรือเรคอร์ดในขณะใดขณะหนึ่ง วิธีที่ง่ายที่สุดในการควบคุมการเขียนข้อมูลลงไฟล์คือ การกำหนดให้ใช้ไฟล์ได้เพียงคนเดียว หรือ Exclusive Use แต่ปัญหาจะเกิดตามมาก็คือ ไม่สามารถแชร์ข้อมูลต่าง ๆ ในแอพพลิเคชันเดียวกันได้ เพราะฉะนั้นเอาอย่างนี้ดีกว่า ให้เปิดไฟล์ในลักษณะ Exclusive Use หรือในลักษณะ "ไฟล์ของข้าใครอย่าแตะ" ในเวลาที่จำเป็นจริง ๆ เท่านั้น

สำหรับโปรแกรมฟอกซ์เบสหรือฟอกซ์โปรจะมีคำสั่งเพียง 6 คำสั่งเท่านั้นที่จะต้องเปิดไฟล์ในลักษณะ EXCLUSIVE USE คือ

- INDEX
- INSERT (BLANK)
- MODIFY STRUCTURE

- PACK
- REINDEX
- ZAP

ถ้าไม่เปิดไฟล์โดย Exclusive Use แล้วมีการใช้คำสั่งทั้ง 6 คำสั่งข้างต้น ก็จะเกิดข้อผิดพลาด (Error) "Exclusive open of file is required"

จะแชร์ข้อมูลกันอย่างไรดี

วิธีการออกแบบระบบให้สามารถแชร์ข้อมูลกันได้นั้น สิ่งหลักที่ต้องคำนึงถึงในการพัฒนาระบบในสภาพแวดล้อมของมัลติยูสเซอร์ ประกอบด้วย 3 หัวข้อใหญ่ คือ

1. ทำอย่างไรจึงแชร์ข้อมูลระหว่างผู้ใช้หรือเวอร์กสเตชันได้
2. ทำอย่างไรจึงจะรักษาความถูกต้องของข้อมูลเอาไว้ได้
3. ทำอย่างไรจึงจะทำให้ทุกคนที่ใช้งานอยู่ในแอพพลิเคชันเดียวกันรู้สึกเสมือนว่าอยู่ในสภาพแบบใช้ข้อมูลอยู่เพียงคนเดียวเท่านั้น

คำสั่งและฟังก์ชันที่ต้องใช้ในระบบมัลติยูสเซอร์

คำสั่ง SET EXCLUSIVE ON/OFF

รูปแบบ

```
SET EXCLUSIVE ON|OFF
```

ตัวอย่าง

```
SET EXCLUSIVE OFF
USE CUSTOMER INDEX ID_CUST
```

เป็นคำสั่งในการกำหนดในกรณีที่ต้องการเปิดไฟล์เพื่อให้ใช้ข้อมูลร่วมกันได้ โดยปกติจะ ON อยู่ คือ ใช้ข้อมูลได้เพียงคนเดียว และถ้ากำหนดให้ OFF หมายถึงให้ Share คือจะใช้ข้อมูลร่วมกันได้

คำสั่ง USE EXCLUSIVE

รูปแบบ

```
USE <filename> INDEX <indexname>
EXCLUSIVE
```

ตัวอย่าง

```
USE CUSTOMER INDEX ID_CUST
EXCLUSIVE
```

จะใช้ไฟล์ CUSTOMER ได้เพียงคนเดียว เช่นเดียวกับการใช้คำสั่ง

```
SET EXCLUSIVE ON
USE CUSTOMER INDEX ID_CUST
```

แต่คำสั่งนี้จะพิเศษอยู่อย่างหนึ่ง คือ จะใช้คู่กับคำสั่ง SET EXCLUSIVE OFF ซึ่งจะเป็นการแชร์ข้อมูล แต่ผู้เขียนโปรแกรมต้องการเปิดไฟล์บางไฟล์แบบ Exclusive Use คือใช้เพียงคนเดียวเพื่อจุดประสงค์บางอย่าง เช่น การสร้างไฟล์ดัชนี (Index File) หรือใช้คำสั่ง ZAP เพื่อล้างไฟล์ให้ว่างเปล่า

ฟังก์ชัน FLOCK( )

เป็นฟังก์ชันสำหรับล็อกไฟล์ที่กำลังเปิดอยู่ เพื่อป้องกันการเข้าถึงข้อมูลจากผู้อื่น ซึ่งจะรีเทอร์นค่า .T. หรือ .F. โดยปกติถ้ายังไม่มีใครล็อกไฟล์จะรีเทอร์นค่า .T. แต่หากมีใครล็อกไฟล์หรือมีการล็อกเรคอร์ดด้วยฟังก์ชัน RLOCK( ) หรือ LOCK( ) แม้เพียงเรคอร์ดด้วย ก็จะรีเทอร์นค่า .F. คือล็อกไฟล์ไม่สำเร็จนั่นเอง

ฟังก์ชัน RLOCK( ) หรือ LOCK( )

จะคล้ายกับฟังก์ชัน FLOCK( ) แต่ RLOCK( ) หรือ LOCK( ) เป็นการล็อกเรคอร์ดที่ต้องการ เพื่อป้องกันการเข้าถึงเรคอร์ดที่ผู้ใช้คนใดคนหนึ่งต้องการจะเปลี่ยนแปลงข้อมูลในเรคอร์ดดังกล่าว หรือป้องกันผู้ใช้คนอื่นมาเปลี่ยนแปลงข้อมูลพร้อมกัน ซึ่งเรียกว่าเกิดการชนกัน (Collision) ได้ โดยจะรีเทอร์นค่า .T. ถ้ายังไม่มีใครล็อกเรคอร์ดนั้น ก็จะทำการล็อกเรคอร์ดนั้นให้ทันที

คำสั่ง UNLOCK

รูปแบบ

```
UNLOCK <ALL>
```

เป็นคำสั่งสำหรับยกเลิกการล็อกไฟล์ที่เกิดจากการใช้ฟังก์ชัน FLOCK( ) หรือยกเลิกการล็อกเรคอร์ดด้วยฟังก์ชัน RLOCK( ) หรือ LOCK( ) เพื่อให้ผู้ใช้คนอื่นสามารถใช้ไฟล์หรือเรคอร์ดนั้น ๆ ต่อไปได้ ถ้าใช้คำสั่ง UNLOCK ALL จะเป็นการยกเลิกการล็อกไฟล์ทั้งหมดที่กำลังเปิดอยู่

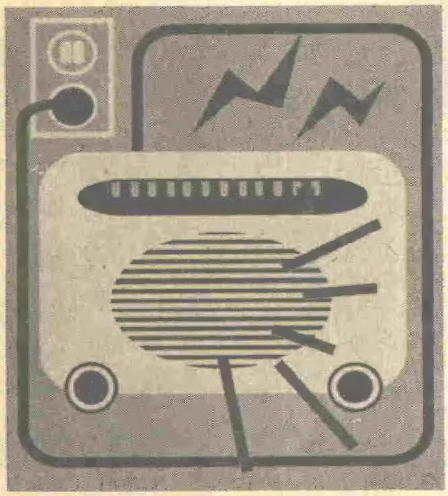
เน็ตเวอร์กแฟล็ก (Network Flags)

ถ้าท่านใช้ระบบเครือข่ายของโนเวลล์เน็ตแวร์จะต้องให้ผู้ดูแลระบบหรือ System Administation กำหนดสิทธิให้กับไดเรคทอรีและไฟล์ต่าง ๆ ที่ต้องการจะแชร์ข้อมูล โดยใช้คำสั่ง FLAG ของเน็ตแวร์ดังนี้

```
FLAG *.* S
```

การล็อกข้อมูลอัตโนมัติ (Automatic Locking)

ในฟอกซ์เบสและฟอกซ์โปรจะมีการล็อกข้อมูลให้อัตโนมัติถ้ามีการใช้คำสั่ง SET LOCK ON เพื่อให้มีการล็อกข้อมูลให้อัตโนมัติ เมื่อมีการใช้คำสั่งดังตารางที่ 1 นี้ หากใช้คำสั่ง SET LOCK OFF จะไม่มีการล็อกข้อมูลอัตโนมัติได้



ข้อผิดพลาด (Error Number) ที่เกิดขึ้นในระบบมัลติยูสเซอร์ แสดงได้ดังตารางที่ 2

ขั้นตอนการเปลี่ยนแปลงโปรแกรมจากผู้ใช้คนเดียวมาสู่ระบบมัลติยูสเซอร์

1. ทำรายการไฟล์ที่ต้องใช้ทั้งหมดในระบบที่จะ Convert

2. ทำรายการ Routine หรือโปรแกรมที่มีการเขียนข้อมูลลงไฟล์หรือมีการเปลี่ยนแปลงข้อมูลในไฟล์

3. ตรวจสอบดูว่าในระบบนั้นมีไฟล์อะไรที่ต้องเปิดแบบ Exclusive บ้าง เช่น การออกรายงานที่ต้องการความถูกต้องของข้อมูลโดยไม่ให้มีการเพิ่มเติม หรือมีการเปลี่ยนแปลงข้อมูลในไฟล์จากผู้ใช้คนอื่นในระบบในขณะที่กำลังออกรายงานอยู่

4. เปิดไฟล์อื่นที่เหลือด้วยการแชร์ข้อมูล โดยใช้คำสั่ง SET EXCLUSIVE OFF

5. ตรวจสอบว่าแต่ละคำสั่งในโปรแกรมซึ่งมีการเขียนข้อมูลลงไฟล์โดยจะต้องมีการล็อกเรคอร์ดเสียก่อนที่จะมีการเขียนข้อมูล และยกเลิกการล็อกข้อมูลเมื่อจบการเขียนข้อมูลแต่ละครั้ง

คำสั่ง	ประเภทการล็อกข้อมูล
APPEND	File
APPEND BLANK	File header
APPEND FROM	File
APPEND FROM ARRAY	File
APPEND MEMO	Record
BROWSE	Record
CHANGE	Record
DELETE	Record
DELETE NEXT 1	Record
DELETE RECORD n	Record
DELETE (NEXT n, ALL, REST)	File
EDIT	Record
GATHER	Record
MODIFY MEMO	Record
READ	Record
RECALL	Record
RECALL NEXT 1	Record
RECALL RECORD n	Record n
RECALL (NEXT n, ALL, REST)	File
REPLACE	Record
REPLACE NEXT 1	Record
REPLACE RECORD n	Record n
REPLACE (NEXT n, ALL, REST)	File
UPDATE	File

ตารางที่ 1 แสดงคำสั่งที่ใช้ในการล็อกข้อมูล

Error No.	Error Message	ความหมาย
108	File is in use by another	มีผู้ใช้อื่นใช้ไฟล์นั้นอยู่
109	Record is in use by another	มีผู้ใช้อื่นใช้เรคอร์ดนั้นอยู่
110	Exclusive open of file is required	จะต้องเปิดไฟล์แบบ Exclusive
111	Cannot write to a read-only file	ไม่สามารถเขียนข้อมูลลงไฟล์ อ่านอย่างเดียว
124	Invalid printer redirection	การเปลี่ยนทิศทางเครื่องพิมพ์ไม่ถูกต้อง
130	Record is not locked	เรคอร์ดยังไม่ได้ล็อก

ตารางที่ 2 แสดงข้อผิดพลาดที่เกิดขึ้นในระบบมัลติยูสเซอร์

6. เพิ่มคำสั่ง ON ERROR routine เพื่อควบคุมการเกิดข้อผิดพลาดในโปรแกรมที่จะเกิดขึ้นจากเน็ตเวอร์ก

7. อย่าลืมว่าจะต้องเพิ่มคำสั่งที่จะตรวจสอบค่า .T. หรือ .F. ที่จะส่งกลับจากการใช้ฟังก์ชัน FLOCK( ),

RLOCK( ), LOCK( )

ปรับปรุงระบบให้ทำงานได้เร็วขึ้น

1. กรณีงานที่ทำกลางคืน (Overnight Update) ให้เปิดไฟล์แบบ Exclusive เพื่อให้ระบบไม่ต้องเสียเวลาในการตรวจสอบการใช้ข้อมูลพร้อมกัน

2. หากโปรแกรมมีการสอบถามข้อมูล หรือพิมพ์รายงานที่ไม่ต้องใช้ไฟล์ดัชนี (Index File) ก็ควรปิดไฟล์ดัชนีที่ไม่ใช้ ทั้งนี้เพื่อลดการใช้หน่วยความจำ และลดการรับส่งข้อมูลภายในเน็ตเวอร์ก

3. ควรเปิดไฟล์เท่าที่จำเป็นในการใช้งานในขณะนั้นเท่านั้น ไม่ควรเปิดไฟล์ที่ไม่จำเป็นทิ้งไว้

4. ควรแนะนำให้ผู้ใช้ Logout ออกจากระบบทุกครั้งที่เลิกใช้งาน ไม่ควร Logon ทิ้งไว้ ทั้งนี้เพื่อให้ระบบเน็ตเวอร์กทำงานได้เร็วขึ้นนั้นเอง

สรุป

ถึงตรงนี้ท่านผู้อ่านคงพอจะได้แนวทางในการพัฒนาระบบงานในลักษณะมัลติยูสเซอร์บ้างไม่มากก็น้อย หากมีข้อซักถามเพิ่มเติมประการใด เขียนจดหมายสอบถามผ่านกองบรรณาธิการมาได้เลย ผู้เขียนยินดีแลกเปลี่ยนความรู้กับท่านเรามาแชร์ความรู้กันบ้างจะเป็นไรไป สำหรับตอนนี้คงจบไว้แค่นี้ ขอบคุณครับ

รูปที่ 1 แสดงตัวอย่างโปรแกรมที่ใช้สร้างเมนู

```
* LAN routine ***************
* By Somsak Thongsart PTT. *
* 25/03/93 ******************

set talk off
set delete on
set procedure to lanrtn
set exclusive off              && Set for share database
on error do error_rtn with error()  && If error goto error_rtn
use customer index id_cust
*** index on cus_no to id_cust
do Main_menu

PROCEDURE Main_menu
      do while .t.
            clear
            @ 1,5 say   '*** MAIN MENU ***'
            @ 3,5 prompt '   1. Add  Record    '
            @ 4,5 prompt '   2. Edit Record    '
            @ 5,5 prompt '   3. Display Record '
            @ 6,5 prompt '   4. Delete record  '
            @ 7,5 prompt '   5. Copy to new file '
            @ 8,5 prompt '   6. Reindex file   '
            @ 9,5 prompt '   0. Quit           '
            menu to choice
            do case
                  case choice = 1
                                 do Add_rec
                  case choice = 2
                                 do Edit_rec
                  case choice = 3
                                 do Disp_rec
                  case choice = 4
                                 do Del_rec
                  case choice = 5
                                 do Copy_fil
                  case choice = 6
                                 do rein_fil
                  otherwise
                              close all
                              return
            endcase
      enddo
RETURN  && Main_menu

PROCEDURE Add_rec
      wcus_no = space(5)
      wcus_name = space(40)
      waddress = space(60)
       @ 11,5 say 'Customer No.  :' get wcus_no
      @ 13,5 say 'Customer Name :' get wcus_name
      @ 15,5 say 'Address       :' get waddress
      read
      *** Check Duplicate record ***
      seek wcus_no
      if found()
            @ 23,10 say 'Duplicate Key, Not add record'
            wait
      else
            append blank
            if rlock()
                  replace cus_no with wcus_no
                  replace cus_name with wcus_name
                  replace address with waddress
                  unlock
            else
                  @ 23,10 say 'Record not locked'
                  wait
            endif
      endif
RETURN  && Add_rec

PROCEDURE Edit_rec
      wcus_no = space(5)
      wcus_name = space(40)
      waddress = space(60)
      @ 11,5 say 'Customer No.  :' get wcus_no
      read
      *** Search data for edit ***
      seek wcus_no
      if      .not. found()
        @ 23,10 say 'Record not found'
          wait
      else
            wcus_name = cus_name
            waddress = address
```

รูปที่ 1 (ต่อ)

```
        @ 13,5 say 'Customer Name :' get wcus_name
        @ 15,5 say 'Address      :' get waddress
        read
        if rlock()
              replace cus_name with wcus_name
              replace address with waddress
              unlock
        else
              @ 23,10 say 'Record in use by another, Not save'
              wait
        endif
    endif
RETURN

PROCEDURE Disp_rec
    wcus_no = space(5)
    wcus_name = space(40)
    waddress = space(60)
    @ 11,5 say 'Customer No.  :' get wcus_no
    read
    *** Search data for display ***
    seek wcus_no
    if .not. found()
       @ 23,10 say 'Record not found'
         wait
    else
         wcus_no = cus_no
         wcus_name = cus_name
         waddress = address
         @ 11,5 say 'Customer No.  :' get wcus_no
         @ 13,5 say 'Customer Name :' get wcus_name
         @ 15,5 say 'Address      :' get waddress
          clear get
         wait
    endif

RETURN  && Disp_rec

PROCEDURE Del_rec
    wcus_no = space(5)
    wcus_name = space(40)
    waddress = space(60)
    @ 11,5 say 'Customer No.  :' get wcus_no
    read
    *** Search data for delete ***
    seek wcus_no
    if .not. found()
       @ 23,10 say 'Record not found'
         wait
    else
         wcus_no = cus_no
         wcus_name = cus_name
         waddress = address
         @ 11,5 say 'Customer No.  :' get wcus_no
         @ 13,5 say 'Customer Name :' get wcus_name
         @ 15,5 say 'Address      :' get waddress
         clear get
        answer = 'N'
        @ 23,10 say 'Confirm Delete Y/N' get answer picture '!'
         read
        if answer = 'Y'
              if rlock()
              delete
              unlock
        else
              @ 24,10 say 'Record in use by another, Not delete'
              wait
        endif
    endif
RETURN  && Del_rec

PROCEDURE Copy_fil
    answer = 'N'
    @ 23,10 say 'Copy file to test.dbf Y/N' get answer picture '!'
    read
    if answer = 'Y'
        if flock()
            copy to test
            unlock
        else
            @ 23,10 say 'File is use by another'
            wait
        endif
    endif
RETURN  && Copy_fil

PROCEDURE Rein_fil
    answer = 'N'
                    @ 23,10 say 'Reindex file Y/N' get answer picture '!'
    read
    if answer = 'Y'
        if flock()
            index on cus_no to id_cust
            unlock
        else
           @ 23,10 say 'File is use by another'
           wait
        endif
    endif
RETURN  && Rein_fil

PROCEDURE Error_rtn
          parameter errnum
          do case
              case errnum = 108
                  @ 23,10 say 'File in use by another'
              case errnum = 109
                  @ 23,10 say 'Record in use by another'
              case errnum = 110
                 @ 23,10 say 'Exclusive open of file is required'
              case errnum = 111
                 @ 23,10 say 'Cannot write to a read-only file'
              case errnum = 124
                 @ 23,10 say 'Invalid printer redirection'
              case errnum = 130
                  @ 23,10 say 'Record is not locked'
              otherwise
                 @ 23,10 say 'ERROR : ' + str(errnum,4) + ;
                                          ' Not network related'
          endcase
          wait
    RETURN  && Error_rtn

    *** End Lanrtn.prg ***
```